4

UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
09/361,782	07/26/1999	BRIAN DEEN	MSI-390US	5941

22801 7590 09/20/2004

LEE & HAYES PLLC
421 W RIVERSIDE AVENUE SUITE 500
SPOKANE, WA 99201

EXAMINER	
QUELER, ADAM M	
ART UNIT	PAPER NUMBER
2179	

DATE MAILED: 09/20/2004

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	**Application No.** 09/361,782	**Applicant(s)** DEEN ET AL.	
	Examiner Adam M Queler	**Art Unit** 2179	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *26 May 2004*.

2a)□ This action is **FINAL**. 2b)☒ This action is non-final.

3)□ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-55* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)□ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-55* is/are rejected.

7)□ Claim(s) _____ is/are objected to.

8)□ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)□ The specification is objected to by the Examiner.

10)□ The drawing(s) filed on _____ is/are: a)□ accepted or b)□ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)□ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)□ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)□ All b)□ Some * c)□ None of:

1.□ Certified copies of the priority documents have been received.

2.□ Certified copies of the priority documents have been received in Application No. _____.

3.□ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)

2) □ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3) □ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08)
Paper No(s)/Mail Date _____.

4) □ Interview Summary (PTO-413)
Paper No(s)/Mail Date. _____ .

5) □ Notice of Informal Patent Application (PTO-152)

6) □ Other: _____.

DETAILED ACTION

1. This action is responsive to communications: Amendment and RCE filed 05/26/2004.

2. Claims 1-55 are pending in the case. Claims 1, 5, 14, 20, 31, 37, 41, and 44 are independent claims.

3. The rejections of the previous claims under 35 U.S.C. § 103 in view of the prior art have been withdrawn.

Continued Examination Under 37 CFR 1.114

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 5/26/2004 has been entered.

Claim Rejections - 35 USC § 103

5. The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

6. **Claims 14, 16, 17, 19, and 37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh et al (USPN 6012098 —2/23/1998) and further in view of Pemburton et al. "XHTML™ 1.0: The Extensible HyperText Markup Language," published 5/5/1995.**

Regarding independent claim 14, Bayeh discloses receiving a request (col. 10, lines 19-25). Bayeh discloses gathering the data (col. 10, lines 46-58). Bayeh teaches calling an emitter object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

As Bayeh makes no mention of building a hierarchical tree, the Office interprets its absence to mean that it also emits data "in a manner in which" a tree would not have to be built. The Office submits that the absence of a tree in Bayeh is sufficient to anticipate the negative limitation. However, for the sake of argument the Bayeh sends the markup language data in data streams (col. 2, ll. 45-49) not trees, that is, the HTML/XML is represented as strings of text, not a hierarchical tree structure.

Regarding dependent claim(s) 16, Bayeh teaches calling multiple emitting objects, which would inherently result in multiple emissions of data (col. 8, ll. 43-49).

Regarding dependent claim(s) 17, Bayeh teaches calling multiple emitting objects, which would inherently result in multiple emissions of data (col. 8, ll. 43-49). Bayeh also teaches a defined order in which servlets are called (col. 9, ll. 47-63)

Regarding dependent claim 19, the program of claim 19 is the program for carrying out the method of claim 14 and is rejected under the same rationale.

Regarding independent claim 37, Bayeh discloses gathering the data (col. 10, lines 46-58) with a servlet, which is an object, which inherently must be called. Bayeh teaches calling an emitter object configured to receive calls that are generated by the data object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12).

Bayeh also teaches that multiple servlets may call multiple servlets (col. 8, ll. 43-45). Bayeh does not explicitly disclose to a pre-defined order of calls. Bayeh also teaches a defined order in which servlets are called (col. 9, ll. 47-63).

Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

7. **Claims 1-7, 10-11, 13, 31-32, 34-35, 38 and 48-51 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh, and further in view of Pemburton, and further in view of "Build Servlet-Based Enterprise Web Applications," by Philion (Copyright 1998).**

Regarding independent claim 1, Bayeh discloses processing and formatting results as HTML (col. 11, line 30 – col. 12, line 12). Preparing is broadly interpreted by the examiner to be equivalent to formatting and processing. Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Bayeh teaches repeating the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach repeating the sending of the prepared portions. Philion teaches sending partial result to a client when it is ready (pp.5-6). In other words, repeating preparing and sending of the response. It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby repeating and preparing and sending until the entire document is sent. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 2, Bayeh discloses gathering the data (col. 10, lines 46-58). Bayeh teaches calling an emitter object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh does not teach formatting into XML, at this step but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Regarding dependent claim 3, Bayeh teaches a gathering object to gather data (col. 10, lines 46-58).

Regarding dependent claim 4, Bayeh discloses receiving a request (col. 10, lines 19-25).

Regarding independent claim 5, Bayeh discloses receiving a request (col. 10, lines 19-25). Bayeh discloses processing and formatting results as HTML (col. 11, line 30 – col. 12, line 12). Preparing is broadly interpreted by the examiner to be equivalent to formatting and processing. Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 6, Bayeh teaches repeating the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach repeating the sending of the prepared portions. Philion teaches sending partial result to a client when it is ready (pp.5-

6). In other words, repeating preparing and sending of the response. It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby repeating and preparing and sending until the entire document is sent. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 7, the claim is rejected similarly as claim 6, and further Bayeh processing a stream (col. 12, ll. 4-12), which is inherently processed in a pre-defined order.

Regarding dependent claim 10, Bayeh discloses gathering the data (col. 10, lines 46-58), with a mechanism, in this case a servlet. Bayeh also teaches formatting data into HTML syntax (col. 1, ll. 1-2), with a mechanism, in this case a servlet.. It would have been obvious in view Pemburton to replace HTML with XHTML for the reasons explained in claim 5 above.

Regarding dependent claim 11, as Bayeh, Pemburton, and Philion disclose sending the response as described in claim 5 above, there was inherently a mechanism to do so.

Regarding dependent claims 13, the program of claim 13 is the program for carrying out the method of claim 5 and is rejected under the same rationale.

Regarding independent claim 31, Bayeh discloses receiving a request (col. 10, lines 19-25). Inherently, Bayeh has a mechanism for dealing with such a request. Bayeh discloses processing and formatting results as HTML (col. 11, line 30 – col. 12, line 12). Preparing is broadly interpreted by the examiner to be equivalent to formatting and processing. Inherently Bayeh has a mechanism for preparing, that is also coupled to request-receiving mechanism as it prepares the response to the said request handled by said request-receiving mechanism. Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3). Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 32, Bayeh teaches repeating the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach repeating the sending of the prepared portions. Philion teaches sending partial result to a client when it is ready (pp.5-6). In other words, repeating preparing and sending of the response. It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby repeating and preparing and sending until the entire document is sent. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 34, the claim is rejected similarly as claim 31, and further Bayeh teaches processing a stream (col. 12, ll. 4-12), which is inherently processed in a pre-defined order.

Regarding dependent claim 35, Bayeh discloses gathering the data (col. 10, lines 46-58). Bayeh also teaches formatting data into XML syntax (col. 11, ll. 1-2).

Regarding dependent claim 38, Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim(s) 48, 49, 50, and 51, Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending before the document is complete. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, sending the response portion before the XML document is completely built.. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

8. **Claims 8, 9, 18, 33, 41-43, 52 and 55 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh, Pemburton and Philion as applied to claims 5, 14 and 31 above, and further in view of "Extensions for Distributed Authoring on the World Wide Web – WebDAV, Internet Draft," by Goland et al (published 4/7/1998).**

Regarding dependent claims 9, 18, and 33, Bayeh, Pemburton and Philion do not disclose a multi-status response. Goland discloses a multi-status response, which is an ordinary XML document (p. 54). It would have been obvious to one of ordinary skill in the art at the time of the invention to use the combination of Bayeh, Pemburton and Philion, which processed XML document as disclosed in claim 5 above, to respond with a multi-status response as it was a normally formatted XML document.

Regarding dependent claim 8, the claim is rejected similarly as claim 7 above. Additionally, Bayeh, Pemburton, and Philion do not disclose a multi-status response. Goland discloses a multi-status response, which is an ordinary XML document (p. 54). It would have been obvious to one of ordinary skill in the art at the time of the invention to use the combination of Bayeh W3C and Heinemann, which processed XML document as disclosed in claim 5 above, to respond with a multi-status response as it was a normally formatted XML document.

Regarding independent claim 41, Bayeh discloses receiving a request (col. 10, lines 19-25). Bayeh teaches gathering data for a response with an object (col. 10, lines 46-58).

Bayeh teaches calling an emitter object configured to receive calls that are generated by the data object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh generates a portion of the response c12.5-13. Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Bayeh teaches there are several types of data servlets, each with a specific task (col. 7, ll. 44-45). Bayeh teaches servlet aliasing, whereby a certain servlet is "invoked" or created based on the task it is needed for (col. 9, ll. 47-64, also col. 8, ll. 50-60), effectively associating the object with the task. Inherent, in choosing the correct servlet would be determining the task. Bayeh, Pemburton and Philion do not disclose the request being a HTTP verb. Goland discloses several WebDAV request methods (ch. 7), the WebDAV methods being HTTP verbs. It would have been obvious to one of ordinary skill in the art at the time of the invention to request XML with WebDAV, thereby replacing the general task of Bayeh and Pemburton, as the specific task of the WebDAV request method, as any data gathering method could be used (Bayeh, col. 10, ll 45-58) and the objects could be optimized for the specific request method (col. 8, ll. 60-63).

Regarding dependent claim 42, Bayeh teaches processing a stream (col. 12, ll. 4-12), which is inherently processed in a pre-defined order.

Regarding dependent claim 43, Bayeh teaches repeating the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach repeating the sending of the prepared portions. Philion teaches sending partial result to a client when it is ready (pp.5-6). In other words, sending the accumulated response. It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby repeating and preparing and sending until the entire document is sent. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim(s) 52, Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending before the document is complete. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, sending the response portion before the XML document is completely built.. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim(s) 55, Bayeh teaches there are several types of data servlets, each with a specific task (col. 7, ll. 44-45). Bayeh teaches servlet aliasing, whereby a certain servlet is "invoked" or created based on the task it is needed for (col. 9, ll. 47-64, also col. 8, ll. 50-60), effectively associating the object with the task. Inherent, in choosing the correct servlet would be determining the task. Bayeh, Pemburton and Philion do not disclose the request being a HTTP verb. Goland discloses several WebDAV request methods (ch. 7), the WebDAV methods being HTTP verbs. It would have been obvious to one of ordinary skill in the art at the time of the invention to request XML with WebDAV, thereby replacing the general task of Bayeh and Pemburton, as the specific task of the WebDAV request method, as any data gathering method could be used (Bayeh, col. 10, ll 45-58) and the objects could be optimized for the specific request method (col. 8, ll. 60-63).

9. **Claims 20-23, 25, 27, 30 44-47 and 54 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh and Pemburton, and further in view of Goland.**

Regarding independent claim 20, Bayeh discloses receiving a request (col. 10, lines 19-25). Bayeh teaches gathering data for a response with an object (col. 10, lines 46-58).

Bayeh teaches calling an emitter object configured to receive calls that are generated by the data object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh generates at least a portion of the response (col. 12, ll. 4-12). Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Bayeh teaches there are several types of data servlets, each with a specific task (col. 7, ll. 44-45). Bayeh teaches servlet aliasing, whereby a certain servlet is "invoked" or created based on the task it is needed for (col. 9, ll. 47-64, also col. 8, ll. 50-60), effectively associating the object with the task. Inherent, in choosing the correct servlet, would be determining the task. Bayeh and Pemburton do not disclose the request being a WebDAV method. Goland discloses several WebDAV request methods (ch. 7). It would have been obvious to one of ordinary skill in the art at the time of the invention to request XML with WebDAV, thereby replacing the general task of Bayeh, as the specific task of the WebDAV request method, as any data gathering method could be used (Bayeh, col. 10, ll 45-58) and the objects could be optimized for the specific request method (col. 8, ll. 60-63).

Regarding dependent claim(s) 21, the recitation of "the response portion" corresponds to "at least a portion of a … response." Bayeh teaches sending at least a portion of the response to the client (col. 12, ll. 13-15).

Regarding dependent claim 22, the Office submits that the absence of a tree in Bayeh is sufficient to anticipate the negative limitation. However, for the sake of argument the Bayeh sends the markup language data in data streams (col. 2, ll. 45-49) not trees, that is, the HTML/XML is represented as strings of text, not a hierarchical tree structure.

Regarding dependent claim 23, Bayeh teaches calling multiple emitting objects, which would inherently result in multiple emissions of data (col. 8, ll. 43-49).

Regarding dependent claim 25, Bayeh teaches calling multiple emitting objects, which would inherently result in multiple emissions of data (col. 8, ll. 43-49). Bayeh also teaches a defined order in which servlets are called (col. 9, ll. 47-63).

Regarding dependent claim(s) 27, Bayeh teaches that portions of the response are created separately, and held until sending (col. 12, ll. 4-12). Inherently, wherever they are stored must be considered a buffer. Bayeh teaches sending the plurality of response portions to the together to the client c12.13-15.

Regarding dependent claim 30, the program for performing the method of claim 20 is rejected under the same rationale.

Regarding dependent claim(s) 54, Bayeh teaches there are several types of data servlets, each with a specific task (col. 7, ll. 44-45). Bayeh teaches servlet aliasing, whereby a certain servlet is "invoked" or created based on the task it is needed for (col. 9, ll. 47-64, also col. 8, ll. 50-60), effectively associating the object with the task. Inherent, in choosing the correct servlet, would be determining the task. Bayeh and Pemburton do not disclose the request being a WebDAV method. Goland discloses several WebDAV request methods (ch. 7). It would have been obvious to one of ordinary skill in the art at the time of the invention to request XML with

WebDAV, thereby replacing the general task of Bayeh, as the specific task of the WebDAV request method, as any data gathering method could be used (Bayeh, col. 10, ll 45-58) and the objects could be optimized for the specific request method (col. 8, ll. 60-63).

Regarding independent claim 44, Bayeh discloses receiving a request (col. 10, lines 19-25). Bayeh teaches gathering data for a response with an object (col. 10, lines 46-58).

Bayeh teaches calling an emitter object configured to receive calls that are generated by the data object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh generates a portion of the response c12.5-13. Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

Bayeh teaches there are several types of data servlets, each with a specific task (col. 7, ll. 44-45). Bayeh teaches servlet aliasing, whereby a certain servlet is "invoked" or created based on the task it is needed for (col. 9, ll. 47-64, also col. 8, ll. 50-60), effectively associating the object with the task. Inherent, in choosing the correct servlet would be determining the task. Bayeh and Pemburton do not disclose the request being a HTTP verb. Goland discloses several WebDAV request methods (ch. 7), the WebDAV methods being HTTP verbs. It would have been obvious to one of ordinary skill in the art at the time of the invention to request XML with WebDAV, thereby replacing the general task of Bayeh and Pemburton, as the specific task of the

WebDAV request method, as any data gathering method could be used (Bayeh, col. 10, ll 45-58) and the objects could be optimized for the specific request method (col. 8, ll. 60-63).

Regarding dependent claim 45, Bayeh teaches that the request is routed to the "proper" object (col. 10, ll. 30-31). The Office interprets this to mean that the object is unique to the request.

Regarding dependent claim 46, Bayeh discloses calling an object (servlet) and passing it the data (col. 11, ll. 20-24).

Regarding dependent claim 47, Bayeh teaches calling an emitter object configured to receive calls that are generated by the data object and format the data contained therein into an appropriate HTML format (col. 11, line 30 – col. 12, line 12). Bayeh generates at least a portion of the response (col. 12, ll. 4-12). Bayeh does not teach sending XML but rather HTML. Pemburton teaches XHTML, a reformulation of HTML as an XML application that is compatible with browsers (p. 2, Abstract). It would have been obvious to replace the HTML of Bayeh with XHTML, because XHTML retains all of HTML's commonly used features, while removing the complexities that makes authoring and design difficult and costly (pp. 4-5, 1.2), as well embracing extensibility and portability (p. 5, 1.3).

10. Claims 24, 26, 28, and 53 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh, Pemburton, and Goland, and further in view of Philion.

Regarding dependent claim 24, Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()"

command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim 26, Bayeh teaches calling multiple emitting objects, which would inherently result in multiple emissions of data (col. 8, ll. 43-49). Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending of only the prepared portion. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, thereby preparing and sending a portion of the response. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim(s) 28, Bayeh and Pemburton do not teach sending less then the whole response. Philion teaches sending less than an entirety of a result to a client (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, thereby repeating and preparing and sending until the entire document is sent. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claim(s) 53, Bayeh teaches the preparation of portions of the response (col. 12, ll. 4-12). However, Bayeh and Pemburton do not teach sending before the document is complete. Philion teaches sending partial result to a client when it is ready (pp.5-6). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, in conjunction with the "flush()" command of Philion, sending the response portion before the XML document is completely built.. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

11. Claims 12, 15, 36, 39, and 40 are rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh, Pemburton, and Philion. Mukhi, "ServerTest" (last modified 8/17/1998) is cited as evidence regarding buffered streams.

Regarding dependent claims 12, 36, and 39, Bayeh and Pemburton do not specifically disclose a buffer. Philion discloses buffering a response portion in a buffered and sending the portion (pp.5-6). Philion does not specifically disclose a threshold can be used. Mukhi is cited as evidence that buffered streams have a threshold, (size[]), and when it is reached the data inside is sent (p. 1, last para.). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, thereby sending the portion when the threshold is reached. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Regarding dependent claims 15 and 40, Bayeh and Pemburton do not specifically disclose a buffer. Philion discloses buffering a response portion in a buffered stream and sending the portion, that is less then a complete response (pp.5-6). Philion does not specifically disclose a threshold can be used. Mukhi is cited as evidence that buffered streams have a threshold, (size[]), and when it is reached the data inside is sent (p. 1, last para.). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, thereby sending the portion when the threshold is reached. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

12. Claim 29 is rejected under 35 U.S.C. 103(a) as being unpatentable over Bayeh, Pemburton and Goland, and further in view of Philion. Mukhi is cited as evidence regarding buffered streams.

Regarding dependent claims 29, Bayeh, Pemburton and Goland do not teach buffered streams. Philion teaches a buffered stream (pp. 5-6). Mukhi is cited as evidence that buffered streams have a threshold, (size[]), and when it is reached the data inside is sent (p. 1, last para.). It would have been obvious to one of ordinary skill in the art at the time of the invention to write the XHTML stream of Bayeh and Pemburton to the PrintWriter buffer of Philion, thereby sending the portion when the threshold is reached. The combination is motivated by the desire to keep the browser active (Philion, p.2, para. 3), therefore creating the illusion to the user that the request is faster (Philion, p. 6, para 2) and to take advantage of the multi-threading capabilities of browsers (Philion, p.6, para. 3).

Response to Arguments

13. Applicant's arguments with respect to the claims have been considered but are moot in view of the new ground(s) of rejection, however will be addressed where still relevant.

Regarding Applicant's remarks on claims 20 and 30:

The majority of arguments for these claims are also moot, as the rejections have been modified. However, Applicant alleges a request method object has not been created. The Office has further explained how Bayeh teaches that objects are created in response to a specific task, which upon the combination with Goland would be the WebDAV request method. See the rejections of claim 10 and 20 above.

Regarding Applicant's remarks on dependent claim(s) 21-22:

Applicant alleges that building a hierarchical tree is a necessary part of XML processing. There are two main ways to deal with XML, and all hierarchical mark up languages. The first involves a tree data structure. The second involves text streams. Trees are not necessary to process streams, as event-based parsers were well known at the time of the invention. The Applicant seems to alleging that the novelty of the invention is that it only sends part of the tree. The Office's position is that no tree at all is taught, nor is necessary. The Office submits that the absence of a tree in Bayeh is sufficient to anticipate the negative limitation. However, for the sake of argument the Bayeh sends the markup language data in text streams (col. 2, ll. 45-49) not trees, that is, the HTML/XML is represented as strings of text, not a hierarchical tree structure.

Regarding Applicant's remarks on 44-47:

Applicant alleges the objects are not created, nor to they correspond to a specific verb. The Office has addressed this its response to the comments of claim 20 above.

Regarding Applicant's remarks on Claims 52-53, intended to be regarding claims 54 and 55:

From the Figure Applicant cited, as well as the use of the word invoked as cited in the rejections above, the Office interprets that Figure 4 is shown as an example of the running invention. The use of the word invoked clearly shows that the object is created when it is invoked.

Conclusion

14. The prior art made of record and not relied upon is considered pertinent to applicant's disclosure. Reilly, "Java Network Programming FAQ"

15. Any inquiry concerning this communication or earlier communications from the examiner should be directed to Adam M Queler whose telephone number is (703) 308-5213. The examiner can normally be reached on Monday-Friday.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Heather R Herndon can be reached on (703) 308-5186. The fax phone number for the organization where this application or proceeding is assigned is 703-872-9306.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free).

AQ

STEPHEN S. HONG
PRIMARY EXAMINER